Eurographics Symposium on Rendering 2010
Jason Lawrence and Marc Stamminger
(Guest Editors)

Volume 29 (2010), Number 4

Fast Estimation and Rendering of Indirect Highlights

J. Laurijssen[1], R. Wang[2], Ph. Dutré[1], and B.J. Brown[1]

[1]Katholieke Universiteit Leuven , Belgium [2]University of Massachusetts Amherst

Abstract

This paper proposes a method for efficiently rendering indirect highlights. Indirect highlights are caused by the primary light source reflecting off two or more glossy surfaces. Accurately simulating such highlights is important to convey the realistic appearance of materials such as chrome and shiny metal. Our method models the glossy BRDF at a surface point as a directional distribution, using a spherical von Mises-Fisher (vMF) distribution. As our main contribution, we merge multiple vMFs into a combined multimodal distribution. This effectively creates a filtered radiance response function, allowing us to efficiently estimate indirect highlights. We demonstrate our method in a near-interactive application for rendering scenes with highly glossy objects. Our results produce realistic reflections under both local and environment lighting.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

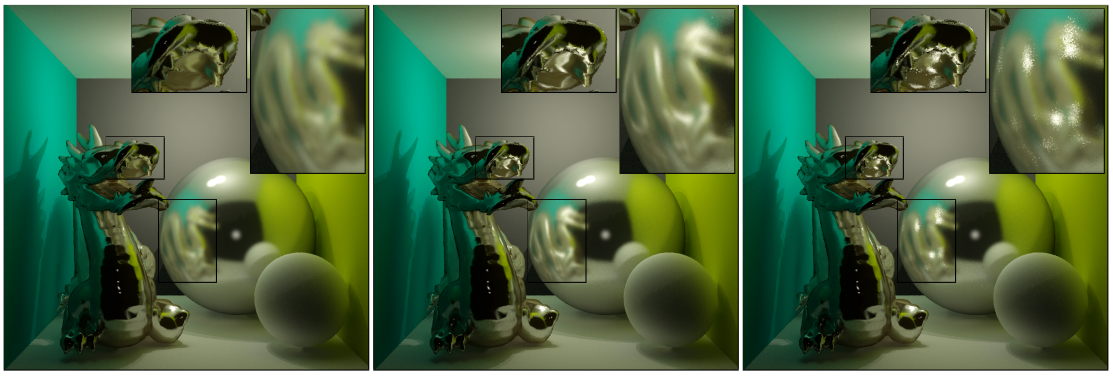
With the rapid improvements in computation power, interactive global illumination has attracted increasing attention in graphics research. Today it is not surprising to see rendering systems that simulate multi-bounce reflections on diffuse and ideally specular surfaces at interactive rates [WWZ*09]. Despite this progress, it remains a challenge to efficiently simulate global illumination effects that fall into the medium-to-high frequency range. Our goal in this paper is to simulate *indirect highlights*, caused by the primary light source reflecting off two or more glossy surfaces (Figure 1). Although indirect highlights are more subtle than diffuse interreflections, they are important to convey the realistic appearance of materials such as chrome and shiny metal.

Consider the extreme case of two non-planar mirror surfaces reflecting light from a point source to the camera. It is non-trivial to find a valid path that connects the light source and camera through reflections on both surfaces, or even whether such a path exists at all. If the surfaces are glossy, many valid paths may exist. But still, only a few make substantial contributions to the reflected radiance, making it hard to find the important paths. In addition, the highly view-dependent nature of such reflections invalidates most caching based schemes [WRC88, Jen01]. Thus in general it takes a long time to sample these paths adequately and compute a noise-free image.

In this paper we treat the glossy BRDF at a surface point as a directional distribution modeled with a spherical von Mises-Fisher (vMF) distribution. As our main contribution, we merge multiple vMFs originating from different surface points into a combined multimodal distribution, effectively creating a filtered radiance response function. This allows us to compute indirect highlights efficiently, without requiring a large number of samples. We demonstrate our technique with a near-interactive application which renders indirect highlights under both local and environment lighting.

2. Related Work

Global illumination has been a long-standing challenge in graphics research. An overview of modern techniques can be found in standard textbooks such as [DBB06]. A fundamental difficulty today is the computation of indirect lighting when many glossy surfaces are present. Standard Monte Carlo methods such as path tracing and Metropolis sampling [Vea97] take a long time to converge. Other techniques such as irradiance caching [WRC88] and photon mapping [Jen01] exploit illumination coherence, but their cost increases rapidly for highly view-dependent effects. Radiance caching [KGPB05, GKBP05, GKB09] caches both spatial and directional samples, which is efficient for moderately glossy surfaces but difficult for highly glossy ones, especially when considering multiple bounces of reflections.



(a) No indirect highlights (b) Our method (c) Path traced

Figure 1: *Our goal is to efficiently render indirect higlights (e.g. LGGE paths [Hec90]). (a) A scene rendered without indirect highlights. (b) The same scene rendered with our method. Indirect highlights clearly contribute to the reflections of the dragon on the sphere. (c) Rendering using a path tracer with the same number of paths traced as (b). Computation times are roughly equal ($\sim$ 3 seconds, at 1024×960 with 64 secondary rays). Our method produces smoother indirect highlights with fewer samples than the path tracer.*

For very glossy materials, it is usually more efficient to trace stochastic rays from the view point. However, because rays diverge upon each reflection, even with proper importance sampling, a large number of sample rays is required to eliminate noise in the image. In this work, our distribution merging method can be seen as an efficient filter applied to stochastic sampling. We effectively create a filtered radiance response function, allowing us to simulate indirect highlights with a greatly reduced number of sample rays.

In [KC08] and [YWY08], illumination filtering is exploited to reduce sampling noise and improve rendering quality; in [RS09], image-space samples are filtered to approximate glossy reflections and other distribution ray tracing effects. These techniques only consider a single bounce of reflections and thus do not produce indirect highlights.

Ray differentials [Ige99] and path differentials [SW01] keep track of each ray's footprint in order to perform filtering during ray tracing. These methods are typically used for texture filtering, but can perhaps be combined with our approach for efficient reflectance filtering.

PRT [SKS02] exploits precomputed transport functions to allow for real-time rendering. Although view-dependent effects are possible with PRT, they are generally expensive to sample and compress. [GKD07] and [WRG*09] approximate BRDFs using Spherical Gaussians (SGs), achieving high-quality results for glossy materials. However, they only consider direct lighting. [GKMD06] and [TS06] use SGs to approximate entire transport functions, which can include both direct and indirect lighting. As with many PRT techniques, they require static geometry and materials.

Our method builds on the von Mises-Fisher (vMF) distribution [HSRG07], which is similar to the Spherical Gaussian except for a normalization factor. Because of the convolution property of vMFs, such distributions have been successfully applied for efficient normal map filtering [Tok05, HSRG07, OB10]. In contrast, we use vMF merging to filter the radiance response function. Our problem is not easily formulated as convolution since our vMF lobes may not be identical.

Another area of related work is based on gathering illumination from many lights [Kel97, WFA*05, HPB07]. These methods assume the source of indirect lighting as many virtual point lights (VPLs). Then, the contribution from each VPL is gathered at every shading point to approximate indirect lighting. Some recent work [Chr08, REG*09] uses many points to represent both the illumination and geometry, and efficiently compute global illumination by projecting these points to microbuffers stored at each shading pixel. These methods exploit the clustering of diffuse point lights, while our method exploits the clustering of reflected BRDF lobes. Thus our method can be combined with existing methods to allow for glossy clustered lights.

Visual spherical lights (VSLs) [HKWB09] extends standard VPLs (which assume diffuse point lights) to include glossy source lights. Though efficient, it currently runs offline, taking several minutes to render a frame. In general, highly glossy materials require many more source point lights, degrading the performance of such an approach.

Recently, Wang et al. [WWZ*09] combine GPU-based photon mapping with adaptive irradiance sampling and photon tree cuts to achieve interactive global illumination. Their method is efficient for diffuse or ideally specular surfaces. On the other hand, simulating high-frequency glossy reflections remains a challenge.

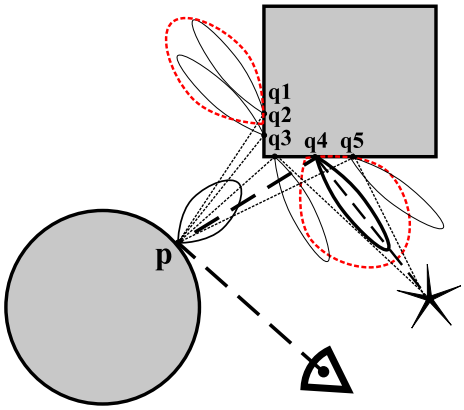


Figure 2: *Indirect highlights are formed by two glossy reflections between the camera and light source. Important paths such as the path through* $\mathbf{q}_4$ *are easily missed if the sampling is not sufficiently dense.*

3. Overview

Figure 2 shows an illustration of indirect highlights, which are created by the primary light reflected off two or more glossy surfaces before reaching the camera. To sample these paths, we can trace paths from the camera to a first surface point $\mathbf{p}$ and perform BRDF importance sampling to continue the path. At each secondary surface point $\mathbf{q}_i$ we send a shadow ray to the light source to obtain the path's contribution from the light source. Due to the narrow support of glossy BRDFs (shown in the figure), important paths are easily missed if we do not sample adequately. This leads to significant noise due to the variance in sampling.

Our solution is to combine the contributions from all points $\mathbf{q}_i$ (in Figure 2) into merged lobes (in red), representing filtered radiance responses. This approximation allows us to quickly compute the total radiance due to indirect highlights, without requiring a large number of sample rays.

Our algorithm can be implemented in any existing path tracer. Specifically, at a secondary intersection $\mathbf{q}_i$, instead of sending a shadow ray to the light source to sample direct lighting, we store each $\mathbf{q}_i$'s position and the corresponding cosine-weighted BRDF lobe. We then merge all BRDF lobes into a small number of clusters and approximate each cluster by a single lobe. We also compute the mean position for each cluster. Finally each merged lobe is used to evaluate direct lighting by sending a shadow ray to the light source. The light source can be an environment map, in which case we use filtered environment maps to return illumination radiance. Because the number of merged lobes is small, we can greatly reduce the required number of shadow rays.

Since we focus on indirect highlights, paths other than indirect highlights are handled using other techniques. In particular, we use standard photon mapping to account for diffuse interreflections. In the following, we first provide a brief review of the vMF distribution; we then discuss the merging algorithm; finally we describe the rendering algorithm.

4. Representation of Directional Distributions

4.1. The von Mises-Fisher Distribution

The von Mises-Fisher (vMF) distribution $\gamma(\mathbf{s})$ [MJ00] is a probability density function that describes the distribution of directions $\mathbf{s}$ centered around a mean direction μ:

$$\gamma(\mathbf{s}) = c(\kappa)e^{\kappa(\mu\cdot\mathbf{s})} \tag{1}$$

where κ is the angular width: higher values of κ correspond to more concentrated distributions. $c(\kappa) = \frac{\kappa}{4\pi\sinh(\kappa)}$ is a normalization factor. Except for $c(\kappa)$, the vMF distribution is the same as a Spherical Gaussian. Assuming $\kappa \gg 1$, the distribution can be approximated as [HSRG07]:

$$\gamma(\mathbf{s}) \approx \frac{\kappa}{2\pi}e^{\kappa(\mu\cdot\mathbf{s}-1)} \tag{2}$$

Suppose that $\mathbf{s}_i$, $i \in [1,M]$ is a set of directions resulting from a stochastic process that can be modeled as a single vMF distribution $\gamma(\mathbf{s})$. We can estimate the parameters μ and κ of this distribution by computing the *unnormalized* average direction $\mathbf{r} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{s}_i$. Intuitively, $\mathbf{r}$ points in the mean direction of the lobe, and the magnitude $\|\mathbf{r}\|$ indicates the angular width of the lobe. For example, a uniform distribution of directions on the sphere results in $\|\mathbf{r}\| = 0$, while an extremely concentrated distribution results in $\|\mathbf{r}\| \approx 1$. Mathematically, unbiased estimators for μ and κ based on $\mathbf{r}$ are given by Banerjee *et al.* [BDGS05] as:

$$\langle\mu\rangle = \frac{\mathbf{r}}{\|\mathbf{r}\|}, \qquad \langle\kappa\rangle \approx \frac{3\|\mathbf{r}\| - \|\mathbf{r}\|^3}{1-\|\mathbf{r}\|^2} \tag{3}$$

4.2. Mixture of vMF Distributions

A vMF distribution describes a single lobe, referred to as a unimodal distribution. In many cases, a set of directions $\mathbf{s}_i$ cannot be adequately described by a single lobe and thus must be modeled as a multimodal distribution. A simple example is a set of directions generated through importance sampling a BRDF with both a diffuse and a glossy component. We represent such a distribution as a mixture of vMFs:

$$\sum_{j=1}^{N}\alpha_j\gamma_j(\mathbf{s}) \tag{4}$$

where α_j is the weight for each lobe, subject to $\sum_{j=1}^{N}\alpha_j = 1$. In order to estimate the parameters of this model, we cluster the set of input directions $\mathbf{s}_i$. For each cluster, vMF parameters are computed (Eq. 3). Banerjee *et al.* [BDGS05] proposed a hard clustering spherical Expectation-Maximization (EM) algorithm, as shown below:

It is an iterative algorithm that starts with N initial vMF estimates. Each iteration consists of an expectation step and a maximization step. The expectation step assigns a sample

Algorithm 4.1: vMF-EM Estimation(S)

repeat
  for each $i \leftarrow 1$ **to** M
    do $\begin{cases} k \leftarrow \text{argmax}_{j'}\, \alpha_{j'} \gamma_{j'}(\mathbf{s}_i) \\ \textbf{for } j \leftarrow 1 \textbf{ to } N \\ \quad \textbf{do } \langle z_{ij} \rangle \leftarrow \begin{cases} 1 & j = k \\ 0 & \text{else} \end{cases} \end{cases}$
  for $j \leftarrow 1$ **to** N
    do $\begin{cases} \langle \alpha_j \rangle \leftarrow \frac{1}{M} \sum_{i=1}^{M} \langle z_{ij} \rangle \\ \langle \mathbf{r}_j \rangle \leftarrow \frac{\sum_{i=1}^{M} \langle z_{ij} \rangle \mathbf{s}_i}{M \langle \alpha_j \rangle} \\ \langle \kappa_j \rangle \leftarrow \frac{3\|\langle \mathbf{r}_j \rangle\| - \|\langle \mathbf{r}_j \rangle\|^3}{1 - \|\langle \mathbf{r}_j \rangle\|^2} \\ \langle \mu_j \rangle \leftarrow \frac{\langle \mathbf{r}_j \rangle}{\|\langle \mathbf{r}_j \rangle\|} \end{cases}$
until *converged*

$\mathbf{s}_i$ to the lobe γ_j that it most likely belongs to. The maximization step then updates the estimated parameters for each lobe. The algorithm typically converges after a small number of iterations. Han *et al.* [HSRG07] show that additional attributes such as color c_j can be added easily to each distribution by adding their estimates to the maximization step.

5. Merging von Mises-Fisher Distributions

As outlined in Section 3, indirect highlights are created by at least two consecutive glossy bounces between the camera and light source (Figure 2). It is very difficult to find significant contributions for these highlights because the probability of a highly glossy BRDF lobe at points $\mathbf{q}_i$ lying in the direction of the light source is small.

Although each individual lobe may be very concentrated, the union of them, represented by the merged distribution, is typically much broader. Thus the probability of the merged distribution lying in the direction of the light source is much higher. This property allows us to estimate indirect highlights more efficiently with greatly reduced sampling noise. Although it is not obvious that the merged distribution should give a correct result, we show in Sections 6.1 and 6.2 that it is indeed a good approximation of indirect highlights.

Below we discuss our method for merging distributions. The input is a large number of unimodal distributions, representing the individual lobes at points $\mathbf{q}_i$. These lobes are computed on the fly, similar to [WRG*09]. Our output is a mixture of vMFs with a small number of lobes representing the merged distributions.

Unimodal Distributions. For now, assume that the merged distribution of a set of vMFs is unimodal and hence can be described by a single vMF (Figure 3(a)). Note that each input vMF can have a different angular width. To estimate such

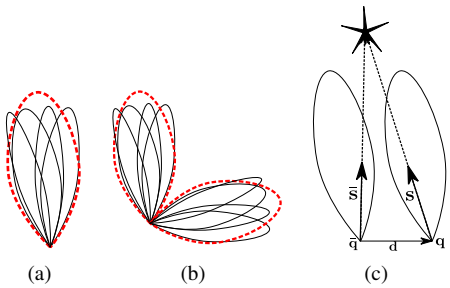


Figure 3: *The total distribution of a set of input vMFs is approximated by (a) a unimodal distribution; and (b) a multimodal distribution. (c) The error introduced by tracing shadow rays from* $\bar{\mathbf{q}}$ *instead of* $\mathbf{q}$ *is a function of the distance* $\mathbf{d}$ *between the two points and the distance* r *of* $\mathbf{q}$ *to the light source. As long as* $\|\mathbf{d}\| \ll r$ *the error is insignificant.*

a distribution, we need to know its unnormalized mean direction $\mathbf{r}_m$, which must be computed from the union of the input vMFs. Since we already know the unnormalized mean direction $\mathbf{r}_i$ of each input vMF, we can exactly compute $\mathbf{r}_m$ by taking a weighted average of $\mathbf{r}_i$:

$$\mathbf{r}_m = \sum_i \alpha_i \mathbf{r}_i \tag{5}$$

where α_i is the relative weight for each input vMF generated during importance sampling. From $\mathbf{r}_m$ we can then estimate the parameters of the merged vMF using Eq. 3.

Multimodal Distributions. As shown in Figure 3(b), the input vMFs can form distinct clusters, therefore in general we must use a multimodal distribution to represent the merged result. To compute individual components of such a mixture model, we cluster $\mathbf{r}_i$ into groups by applying the EM estimation (Algorithm 4.1) on $\frac{\mathbf{r}_i}{\|\mathbf{r}_i\|}$. Each of these clusters is then used to create one vMF distribution according to Eq. 5.

Discussion. In case each input vMF $\gamma(s)$ has equal κ (i.e. angular width), the merging can be formulated as a convolution problem. Specifically, we can compute the distribution $D(s)$ of the lobe centers μ_i and the merged vMF is simply the convolution of $\gamma(s)$ with $D(s)$. This approach has been studied in the context of normal map filtering by [HSRG07, Tok05]. However in our case, the input vMFs do not have equal κ, therefore we cannot directly use the convolution formula.

In theory our lobe merging algorithm can be problematic when the input lobes have very different angular widths but lobe centers that are close to each other. In practice this is seldom an issue, mainly because we are only concerned with highly glossy BRDFs whose lobes are sharp. Wide lobes (i.e. diffuse components of the BRDF) are processed separately. If we look at the relationship between κ and $\|\mathbf{r}\|$ in Eq. 3, we can see that a significant range of the κ value corresponds to a very narrow range of $\|\mathbf{r}\|$ at $\|\mathbf{r}\| \approx 1$. This means that the span in directions among the input lobes, which is likely to

be wider than each individual lobe, has a much stronger influence on the width of the merged lobe. Therefore, in practice merging lobes with different angular widths does not pose any serious problem.

Comparisons to LightCuts. Our method bears some similarity to the LightCuts algorithm [WFA*05], which clusters many diffuse VPLs based on their positions and orientations in a hierarchical manner. Each cluster in the hierarchy contains a VPL representative that exemplifies all VPLs in the cluster. The main difference of our method is that we do not use a VPL-based approach, instead, we use ray tracing to find secondary intersection points. In addition, our method aims to cluster and merge reflected BRDF lobes, while LightCuts aim to cluster diffuse VPLs. Our method can in fact be used as an extension in LightCuts to enable glossy (non-diffuse) VPLs.

6. Applications

In this section we describe how the lobe merging algorithm is used in our rendering applications. We also discuss the approximation accuracy and details of these applications.

6.1. Theory

Light transport through a scene is described by the rendering equation [Kaj86], which states that the radiance $L(\mathbf{p} \rightarrow \mathbf{s}_o)$ from point $\mathbf{p}$ in direction $\mathbf{s}_o$ is the sum of the emitted radiance $L_e(\mathbf{p} \rightarrow \mathbf{s}_o)$ and the reflected radiance:

$$L(\mathbf{p} \rightarrow \mathbf{s}_o) = L_e(\mathbf{p} \rightarrow \mathbf{s}_o) + \int_{\mathbf{S}} \tilde{\rho}(\mathbf{p},\mathbf{s},\mathbf{s}_o) L(\mathbf{p} \leftarrow \mathbf{s}) d\omega_{\mathbf{s}} \quad (6)$$

where $\tilde{\rho}(\mathbf{p},\mathbf{s},\mathbf{s}_o)$ is the cosine-weighed BRDF at surface point $\mathbf{p}$. In the following we focus on indirect highlights formed by two bounces of glossy reflections. Defining secondary surface points $\mathbf{q}$ such that $L(\mathbf{p} \leftarrow \mathbf{s}) = L(\mathbf{q} \rightarrow -\mathbf{s})$, we can expand the rendering equation with one additional level of recursion:

$$L_1(\mathbf{p} \rightarrow \mathbf{s}_o) = \int_{\mathbf{S}} \int_{\mathbf{S}'} \tilde{\rho}(\mathbf{p},\mathbf{s},\mathbf{s}_o) \tilde{\rho}(\mathbf{q},\mathbf{s}',-\mathbf{s}) L_e(\mathbf{q} \leftarrow \mathbf{s}') d\omega_{\mathbf{s}'} d\omega_{\mathbf{s}} \quad (7)$$

We assume that the illumination $L_e(\mathbf{q} \leftarrow \mathbf{s}')$ is distant relative to $\mathbf{q}$, thus it does not change much for different points $\mathbf{q}$, as long as these points are close to each other. We choose a mean point $\bar{\mathbf{q}}$ with illumination $L_e(\bar{\mathbf{q}} \leftarrow \mathbf{s}')$ to be the representative of all points $\mathbf{q}_i$ in a local area. The error introduced by this assumption is discussed later. Eq. 7 can now be reordered as follows:

$$L_1(\mathbf{p} \rightarrow \mathbf{s}_o) = \int_{\mathbf{S}'} L_e(\bar{\mathbf{q}} \leftarrow \mathbf{s}') \int_{\mathbf{S}} \tilde{\rho}(\mathbf{p},\mathbf{s},\mathbf{s}_o) \tilde{\rho}(\mathbf{q},\mathbf{s}',-\mathbf{s}) d\omega_{\mathbf{s}} d\omega_{\mathbf{s}'} \quad (8)$$

We are particularly interested in the inner integral, which can be seen as an effective BRDF computed as:

$$\tilde{\rho}(\bar{\mathbf{q}},\mathbf{s}',\mathbf{s}_o) = \int_{\mathbf{S}} \tilde{\rho}(\mathbf{p},\mathbf{s},\mathbf{s}_o) \tilde{\rho}(\mathbf{q},\mathbf{s}',-\mathbf{s}) d\omega_{\mathbf{s}} \quad (9)$$

Here $\tilde{\rho}(\bar{\mathbf{q}},\mathbf{s}',\mathbf{s}_o)$ expresses the combined contribution of the BRDF lobes $\tilde{\rho}(\mathbf{q},\mathbf{s}',-\mathbf{s})$ to the indirect highlight. Our algorithm estimates $\tilde{\rho}(\bar{\mathbf{q}},\mathbf{s}',\mathbf{s}_o)$ with vMF merging (Section 5), and then evaluates Eq. 8 to obtain the total indirect highlights.

6.2. Algorithm

We start from a shading point seen by the camera. For each such shading point, we trace secondary rays distributed according to the BRDF at the shading point $\tilde{\rho}(\mathbf{p},\mathbf{s},\mathbf{s}_o)$. For each secondary intersection point we store the point's location $\mathbf{q}_i$ and the vMF distribution γ_i corresponding to $\tilde{\rho}(\mathbf{q}_i,\mathbf{s}',-\mathbf{s})$. Because a spherical probability density function such as the vMF distribution integrates to 1, we also need to store an associated color c_i for each γ_i which corresponds to $\int_{\mathbf{S}'} \tilde{\rho}(\mathbf{q}_i,\mathbf{s}',-\mathbf{s}) d\omega_{\mathbf{s}'}$, attenuated with the color of the BRDF at the shading point $\mathbf{p}$.

We proceed by merging the set of distributions γ_i. This results in a multimodal estimate of the distribution corresponding to $\tilde{\rho}(\bar{\mathbf{q}},\mathbf{s}',\mathbf{s}_o)$. We need to make sure that the total integrand of this mixture corresponds to $\int_{\mathbf{S}'} \tilde{\rho}(\bar{\mathbf{q}},\mathbf{s}',\mathbf{s}_o) d\omega_{\mathbf{s}'}$, which we achieve by scaling by a constant a such that

$$a \sum_{j=1}^{M} \alpha_j c_j = \frac{1}{N} \sum_{i=1}^{N} c_i \quad (10)$$

We also compute for each cluster j a position $\bar{\mathbf{q}}_j$ as the mean position of all its cluster members. Finally we perform a direct lighting computation for each of the lobes $a\alpha_j c_j \gamma_j(\mathbf{s}')$ at position $\bar{\mathbf{q}}_j$.

Note that since the mean position $\bar{\mathbf{q}}_j$ may lie underneath the surface, the direct lighting computation may incorrectly return zero due to immediate intersection with the surface. We avoid this by choosing an appropriate ε value for each shadow ray. Alternatively, we could pick a representative point from all the secondary intersection points, instead of using their mean position.

6.3. Approximation Error

We assume that the incident lighting and visibility do not vary much among the secondary intersection points $\mathbf{q}$. This assumption allows us to trace shadow rays starting from a mean position $\bar{\mathbf{q}}$ for all $\mathbf{q}_i$'s. However this approximation may introduce error, if the light source is not distant from all $\mathbf{q}_i$'s. To see this, consider a scene with two glossy reflectors and a point light source which emits radiance L uniformly. For each point $\mathbf{q}$, the error is

$$\varepsilon = c(\kappa)(e^{\kappa\mu\bar{s}} - e^{\kappa\mu s}) = c(\kappa)e^{\kappa\mu s}(1 - e^{\frac{\mu d}{r}})L \quad (11)$$

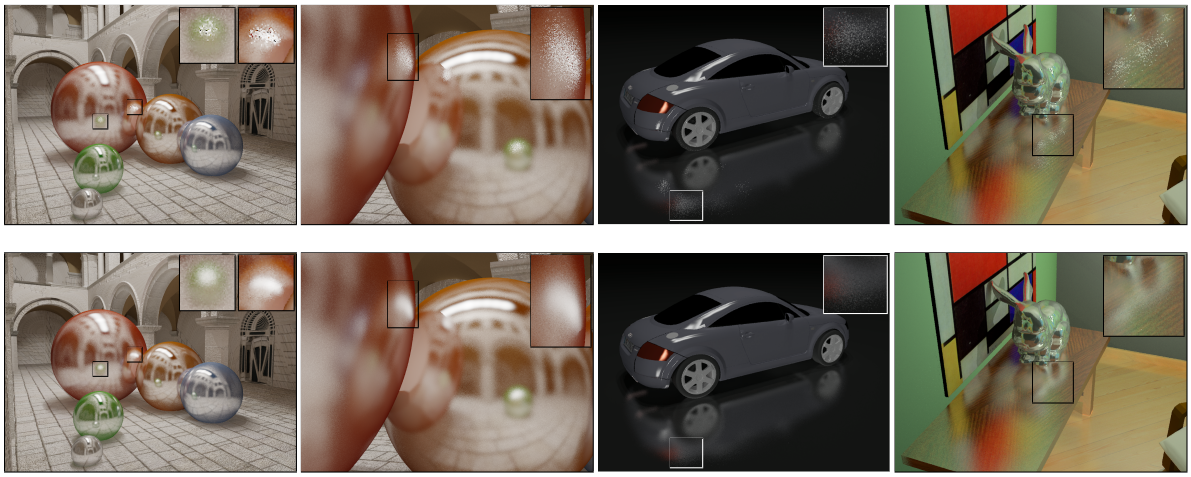


Figure 4: *Top row: indirect highlights rendered with a standard path tracer. Bottom row: our results computed with the same number of samples. Our method produces smoother indirect highlights at roughly the same rendering time.*

with $\mathbf{d} = \mathbf{q} - \bar{\mathbf{q}}$ and r the distance to the light source (Figure 3c). Thus to keep the error low, the distance of each point to the mean position $\bar{\mathbf{q}}$ should be relatively small compared to the distance to the light source. For environment lighting, the light source is distant by definition; for local lighting, we found this assumption to be true in most cases.

An addition source of error is the variation of shadow ray visibility from each point $\mathbf{q}$ to the light source. Even with environment lighting, the visibility may vary between the $\mathbf{q}$'s. However, in practice this is seldom an issue, because all points belonging to the same cluster are fairly concentrated spatially, so the visibility variation is typically very small.

6.4. Environment Lighting

Our approach can also render scenes lit by environment lighting. Some simplifications can be made in this case. First of all we use vMF prefiltered environment maps similar to those used by Green *et al.* [GKD07]. Furthermore instead of evaluating visibility after merging, we decide whether to add a lobe at point $\mathbf{q}_i$ based on one shadow ray in the mean direction μ of that lobe. Finally we do not need to compute mean positions $\bar{\mathbf{q}}_j$ for each cluster, since the direction of lighting is independent of the point position.

7. Results

Implementation. We have implemented our algorithm in a GPU renderer using NVidia's OptiX [PBD*10]. The implementation runs on an Intel Core2 Quad Q9450 2.66GHz processor with an NVIDIA Quadro FX5800 graphics card. Both local lighting and environment lighting are supported.

In the local lighting case, we assume one or a few primary point light sources. We use a GPU-based photon mapper to account for multi-bounce diffuse indirect lighting. To do so, we start from the photon mapper included in the OptiX SDK, which runs on the GPU and is updated every time the lighting changes. To accelerate final gathering, we add the photon irradiance cache as introduced in [Chr99]. This calculates irradiance ahead of time for a small number of selected photons, using density estimation. The current photon mapping implementation we use takes less than a second to re-build when the light source changes. In future work we will integrate state-of-the-art GPU-based photon mapping [ZHWG08] to further improve the speed of this step.

To render, at each primary intersection point $\mathbf{p}$ we compute direct lighting and trace one diffuse final gather sample. These samples are then filtered in image space with a bilateral filter [TM98], resulting in smooth diffuse indirect lighting. Note that we use the bilateral filter only for smoothing the results, not for upsampling. Typically we use a kernel with a width of 30-40 pixels. Next, we trace additional secondary rays, importance sampled according to the glossy BRDF. At each secondary intersection point $\mathbf{q}_i$ we store its corresponding vMF lobe for vMF merging; we also trace a final gather sample for both the diffuse and glossy component of the BRDF.

In the environment lighting case, we do not use the photon mapper to account for diffuse indirect lighting, due to its cost. Therefore our diffuse reflections in this case only include direct illumination. This is a reasonable simplification since diffuse indirect lighting is often not significant in environment lighting. However, we must account for shadows or the rendering would look overly bright. To include global shadowing effects, we compute an ambient occlusion lobe at each vertex on the GPU before the rendering starts. The lobe is computed from unoccluded shadow rays sampled from each vertex. This step takes no more than a few seconds, and is computed only once at the start.

(a) without merging (b) with merging

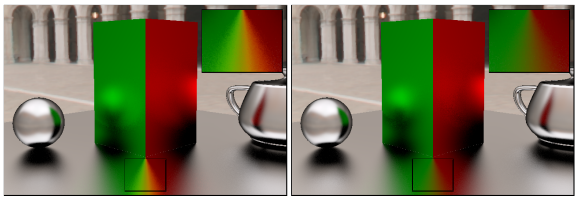
Figure 5: *The left image is rendered without vMF merging. Each individual lobe in points* $\mathbf{q}_i$ *is used to index the pre-filtered environment map. The right image is rendered with vMF merging, which produces much smoother results.*

Rendering Quality. In all our scenes we use Phong specular BRDFs with exponents 64–512. We also include small diffuse components to the BRDFs to increase the rendering realism. We first compare the results of our method against a standard path tracer. We are interested to see if our algorithm renders indirect highlights better for a given sampling budget. In Figure 4 we compare both algorithms for some scenes. Both implementations are identical except for the indirect highlights component: the top row does not apply vMF merging. Each individual lobe at the secondary intersection point requests a shadow ray to obtain illumination radiance. The bottom row shows our algorithm. The rendering times are roughly the same for both algorithms. Our results are smoother with less noise.

This same observation holds for the environment lighting case shown in Figure 5. We notice that merging allows us to render high-frequency lighting contributions (such as the bright lights in Grace Cathedral) much more efficiently.

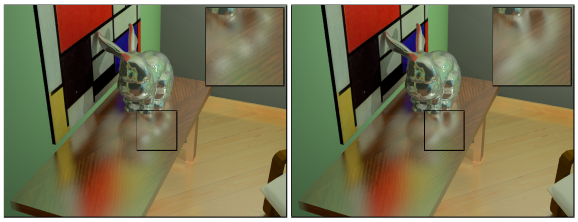
During experiments, we found that in most cases the merged distribution is well approximated by a unimodal distribution. We could argue whether it is necessary to consider a multimodal distribution. The answer is yes – the artifacts introduced by using a unimodal-only representation can be very severe sometimes. As an example, in Figure 6 we show three colored perpendicular glossy planes. In the bottom plane the reflections of the two vertical planes are visible. Around the reflection of the edge shared by the two vertical planes, the merged distribution contains contributions from both planes. Without clustering (i.e. using unimodal only), these contributions are merged into a nearly diffuse distribution, and artifacts are clearly visible (Figure 6a). With clustering enabled, the algorithm automatically finds a multimodal distribution that consists of two clusters. Therefore the artifacts are eliminated (Figure 6b).

On the other hand, we found that in places where the reflector contains rapidly varying high curvature, such as the bunny example in Figure 7a–b, our clustering algorithm may fail to detect clusters and thus incorrectly return a unimodal distribution. This leads to indirect highlights which are too smooth. However, compared to standard path tracing, which



(a) without clustering (b) with clustering

Figure 6: *The reflection of the edge between the two colored vertical planes contains contributions from both planes. (a) no clustering is applied, leading to artifacts in the reflections. (b) clustering is enabled, and artifacts are eliminated.*



(a) Path traced (b) Our method

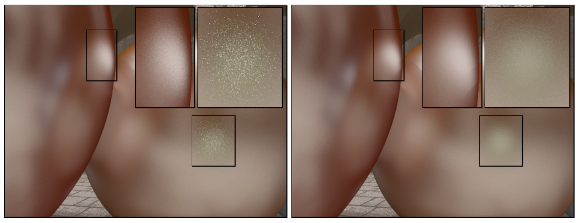
Figure 7: *Clustering for surfaces reflecting a surface with high-frequency variation in curvature sometimes produces a unimodal distribution resulting in too smooth indirect highlights (b) compared to ground truth (a).*

typically produces noisy artifacts (shown in Figure 4 top row), these smooth highlights are often preferred.

Performance. Table 1 shows the performance of our algorithm. All images are rendered at 640×480 resolution. We trace 32 secondary rays and merge lobes into multimodal distributions with two vMFs. Each image is rendered in just a few seconds. Table 1 also shows the percentage cost of each major rendering step, including photon mapping, direct lighting, indirect diffuse, tracing secondary rays, and clustering. The performance varies depending on the number of secondary rays that need to be traced and whether or not these rays hit a surface.

The most challenging scenes are the sponza and bunny scenes where many glossy surfaces are present, and most secondary rays hit some surface. On the other hand, scenes with environment lighting are much faster to render, often at interactive frame rates. In scenes with a mix of diffuse and glossy surfaces such as the dragon scene, computing the diffuse indirect lighting (which is not the main focus of our algorithm) is the main computation bottleneck. The overall cost of clustering is very small. Thus we pay only a very low price to avoid artifacts as those shown in Figure 6.

Glossiness Figure 8 shows how the glossiness of the reflectors influence the rendering quality of our approach. The sphere on the right has medium glossiness (Phong exponent $e = 32$) while two other spheres (one on the left, and one



(a) path traced (b) vMF merging

Figure 8: *The effect of glossiness on the rendering quality. In this scene the sphere on the right has a Phong specular exponent* 32*. The sphere on the left and the green sphere visible in the reflection both have a Phong specular exponent* 512*. Our algorithm improves the rendering quality of the reflected green sphere, which is difficult to sample due to the highly glossy BRDF on that sphere.*

Scene	L_e	# F	P	D	I	S	C	Fps
Sponza	L	66K	2.1	2.7	10.1	80.1	5.0	0.13
Car	L	371K	6.4	11.8	77.3	4.4	0.1	0.84
Bunny	L	72K	1.5	1.3	10.3	78.5	8.4	0.14
Dragon	L	20K	11.1	12.2	71.2	3.6	1.9	1.01
Glasses	E	21K	-	32.5	-	44.0	23.5	2.22
Teapot	E	4K	-	57.7	-	32.3	10.0	4.27

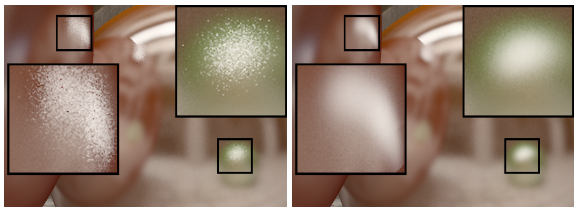
Table 1: *Performance of our GPU renderer. All images are* 640×480*. The second column lists whether the scene is lit by environment lighting (E) or local lighting (L). The third column shows the number of faces of each scene. Columns 4–8 list the percentage (%) cost of the following rendering steps: photon mapping (P), direct lighting (D), indirect diffuse (I), tracing secondary rays (S) and clustering (C). The last column shows the frame rates.*

green sphere visible through reflections) are highly glossy ($e = 512$). The reflection of the green sphere is a difficult case for path tracing because it contains important paths that are difficult to sample. In particular, the highly glossy BRDF on the green sphere leads to narrow reflected lobes at the secondary intersection points. This is the case described in Figure 2. Our algorithm greatly improves the rendering quality in such cases, due to the filtering of reflected lobes. On the other hand, we do not improve the reflection of the medium glossy sphere significantly, since such cases are relatively easy to compute with path tracing.

In Figure 9 we show an example where we set all spheres to use Phong specular BRDF with exponent $e = 512$. Again, our results are much smoother in areas of indirect highlights.

8. Conclusion

We have presented a technique to merge multiple directional distributions, which allows us to compute indirect highlights efficiently. We demonstrated our technique under both local lighting and environment lighting, and achieved near-



(a) path traced (b) vMF merging

Figure 9: *In this example, we set all spheres to use Phong specular BRDF with exponent 512. The left image is rendered without vMF merging, while the right image is rendered with vMF merging.*

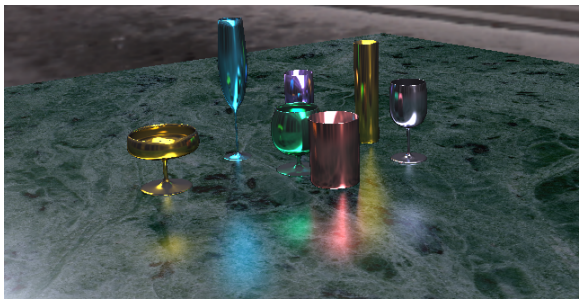


Figure 10: *The Glasses scene illuminated by environment lighting. The rendering speed is 2.22 fps.*

interactive rendering speed using a GPU implementation without any precomputation. As correct simulation of indirect highlights is important to convey the appearance of glossy materials, we believe that our technique is a valuable complement to existing global illumination methods.

For future work we would like to examine our technique when using more than two bounces of glossy reflections. It would be interesting to study the effect of using our merged lobes for additional sampling. We also plan to investigate how to extend our technique to anisotropic materials e.g. by using the Fisher-Bingham distribution [MJ00].

Acknowledgments We would like to thank the anonymous reviewers for their thoughtful comments, and Toon Lenaerts and Roeland Schoukens for modeling the scenes. Jurgen Laurijssen is supported by grant CREA/08/017 from the Research Fund K.U.Leuven (Onderzoeksfonds K.U.Leuven); Rui Wang is supported in part by NSF CAREER grant CCF-0746577. Benedict J. Brown is a post-doctoral fellow of the Research Foundation - Flanders (FWO).

References

[BDGS05] BANERJEE A., DHILLON I. S., GHOSH J., SRA S.: Clustering on the unit hypersphere using von mises-fisher distributions. *J. Mach. Learn. Res. 6* (2005), 1345–1382.

[Chr99] CHRISTENSEN P. H.: Faster photon map global illumination. *Journal of Graphics Tools 4*, 3 (1999), 1–10.

[Chr08] CHRISTENSEN P.: *Point-based Approximate Color Bleeding*. Tech. rep., Pixar Technical Memo #08-01, 2008.

[DBB06] DUTRÉ P., BALA K., BEKAERT P.: *Advanced Global Illumination, second edition*. A. K. Peters, Ltd., 2006.

[GKB09] GASSENBAUER V., KŘIVÁNEK J., BOUATOUCH K.: Spatial directional radiance caching. *Computer Graphics Forum 28*, 4 (2009), 1189–1198.

[GKBP05] GAUTRON P., KŘIVÁNEK J., BOUATOUCH K., PATTANAIK S. N.: Radiance cache splatting: A gpu-friendly global illumination algorithm. In *Proc. EGSR* (2005), pp. 55–64.

[GKD07] GREEN P., KAUTZ J., DURAND F.: Efficient reflectance and visibility approximations for environment map rendering. *Computer Graphics Forum 26*, 3 (2007), 495–502.

[GKMD06] GREEN P., KAUTZ J., MATUSIK W., DURAND F.: View-dependent precomputed light transport using nonlinear gaussian function approximations. In *Proc. I3D* (2006), pp. 7–14.

[Hec90] HECKBERT P. S.: Adaptive radiosity textures for bidirectional ray tracing. In *Proc. SIGGRAPH* (1990), pp. 145–154.

[HKWB09] HAŠAN M., KŘIVÁNEK J., WALTER B., BALA K.: Virtual spherical lights for many-light rendering of glossy scenes. *ACM Trans. Graph. 28*, 5 (2009), 1–6.

[HPB07] HAŠAN M., PELLACINI F., BALA K.: Matrix row-column sampling for the many-light problem. *ACM Trans. Graph. 26*, 3 (2007), 26.

[HSRG07] HAN C., SUN B., RAMAMOORTHI R., GRINSPUN E.: Frequency domain normal map filtering. *ACM Trans. Graph. 26*, 3 (2007), 28.

[Ige99] IGEHY H.: Tracing ray differentials. In *Proc. SIGGRAPH* (1999), pp. 179–186.

[Jen01] JENSEN H. W.: *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., 2001.

[Kaj86] KAJIYA J. T.: The rendering equation. In *Proc. SIGGRAPH* (1986), pp. 143–150.

[KC08] KRIVÁNEK J., COLBERT M.: Real-time shading with filtered importance sampling. *Computer Graphics Forum 27*, 4 (2008), 1147–1154.

[Kel97] KELLER A.: Instant radiosity. In *Proc. SIGGRAPH* (1997), pp. 49–56.

[KGPB05] KŘIVÁNEK J., GAUTRON P., PATTANAIK S., BOUATOUCH K.: Radiance caching for efficient global illumination computation. *IEEE TVCG 11*, 5 (2005), 550–561.

[MJ00] MARDIA K. V., JUPP P. E.: *Directional Statistics*. John Wiley and Sons Ltd., 2000.

[OB10] OLANO M., BAKER D.: Lean mapping. In *Proc. I3D* (2010), pp. 181–188.

[PBD*10] PARKER S. G., BIGLER J., DIETRICH A., FRIEDRICH H., HOBEROCK J., LUEBKE D., MCALLISTER D., MCGUIRE M., MORLEY K., ROBISON A., STICH M.: Optix: A general purpose ray tracing engine. *ACM Transactions on Graphics* (August 2010).

[REG*09] RITSCHEL T., ENGELHARDT T., GROSCH T., SEIDEL H.-P., KAUTZ J., DACHSBACHER C.: Micro-rendering for scalable, parallel final gathering. *ACM Trans. Graph. 28*, 5 (2009), 1–8.

[RS09] ROBISON A., SHIRLEY P.: Image space gathering. In *Proc. HPG* (2009), pp. 91–98.

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph. 21*, 3 (2002), 527–536.

[SW01] SUYKENS F., WILLEMS Y. D.: Path differentials and applications. In *Proc. EGRW* (2001), pp. 257–268.

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *ICCV* (1998), pp. 839–846.

[Tok05] TOKSVIG M.: Mipmapping normal maps. *Journal of Graphics Tools 10*, 3 (2005), 65–71.

[TS06] TSAI Y.-T., SHIH Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph. 25*, 3 (2006), 967–976.

[Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997.

[WFA*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: a scalable approach to illumination. *ACM Trans. Graph. 24*, 3 (2005), 1098–1107.

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. In *Proc. SIGGRAPH* (1988), pp. 85–92.

[WRG*09] WANG J., REN P., GONG M., SNYDER J., GUO B.: All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Trans. Graph. 28*, 5 (2009), 1–10.

[WWZ*09] WANG R., WANG R., ZHOU K., PAN M., BAO H.: An efficient gpu-based approach for interactive global illumination. *ACM Trans. Graph. 28*, 3 (2009), 1–8.

[YWY08] YU X., WANG R., YU J.: Interactive glossy reflections using gpu-based ray tracing with adaptive lod. *Comput. Graph. Forum 27*, 7 (2008), 1987–1996.

[ZHWG08] ZHOU K., HOU Q., WANG R., GUO B.: Real-time kd-tree construction on graphics hardware. *ACM Trans. Graph. 27*, 5 (2008), 1–11.